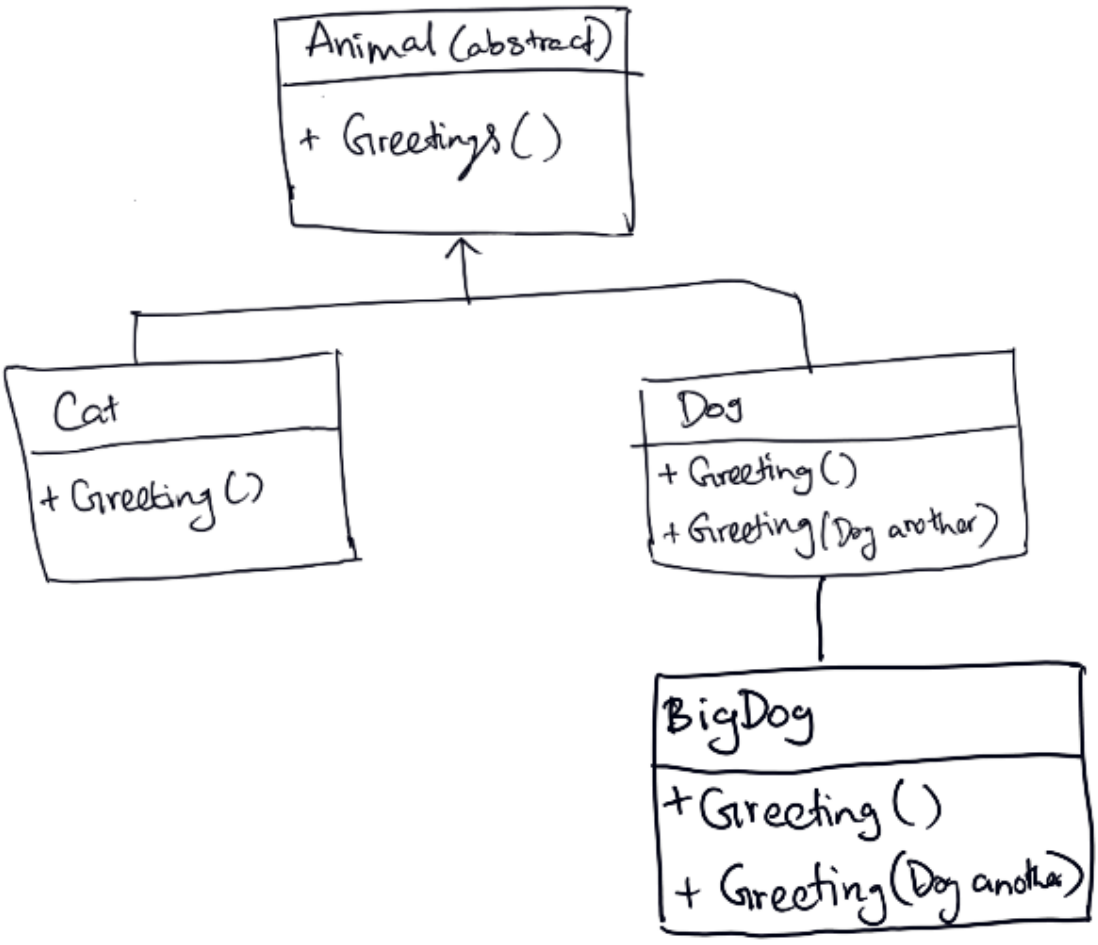
SIT232 Object Oriented Development

By Venujan Malaiyandi
BSCP|CS|61|101

Task 6.1P
Question 4



Question 5

Using the subclasses

Cat cat1 = new cat();

This line creates a new instance of the CAT class and assigns it to the variable cat1.

cat1.greeting();

Calls the greeting() method of the Cat class.

Dog dog1 = new Dog();

Creates a new instance of the Dog class and assigns it to the variable dog1

dog1.greeting();

Calls the greeting() method of the Dog class.

BigDog bigDog1 = new BigDog();

Creates a new instance of the BigDog class and assigns it to the variable bigDog1.

bigDog1.greeting();

Calls the greeting() method of the BigDog class.

Using Polymorphism

Animal animal1 = new Cat();

Creates a new instance of the Cat class and assigns it to the animal reference class, of variable animal1.

animal1.greeting();

Calls the greeting() method of the Cat class, due to the polymorphism.

Animal animal2 = new Dog();

Creates a new instance of the Dog class and assigns it to the variable animal2.

animal2.greeting();

Calls the greeting() method of the Dog class, due to the polymorphism.

Animal animal3 = new BigDog();

Creates a new instance of the BigDog class and assigns it to the variable animal3.

Animal3.greeting();

Calls the greeting() method of the BigDog class, due to the polymorphism.

Animal animal4 = new Animal();

This line would cause an **error**, as you cannot directly instantiate an abstract class.

DownCast

Dog dog2 = (Dog)animal2;

Attempts to downcast animal2. This will succeed as animal2 was originally a Dog.

BigDog bigDog2 = (BigDog)animal3;

This line will downcast animal3.

Dog dog3 = (Dog)animal3;

InvalidCastException will occur as the downcast animal3 to a Dog

Cat cat2 = (Cat)animal2;

This line will attempt to downcast animal2 to a Cat. This will throw an "InvalidCastException" at runtime because directly downcasting an object to a type which its not is not possible.

dog2.greeting(dog3);

This line will call the greeting method of Dog class, passing dog3 (as Dog).

dog3.greeting(dog2);

This line will cause an **error**.

dog2.greeting(bigDog2);

This line will call the greeting method of Dog class, passing bigDog2. This will call the greeting method of Dog class.

bigDog2.greeting(dog2);

This line will call the greeting method of BigDog class, passing dog2.

bigDog.greeting(bigDog1);

This will call the greeting method of BigDog class.